IEEE Copyright Statement:

Progressive 3D Reconstruction of Unknown Objects using one Eye-in-Hand Camera

Guillaume Walck and Michel Drouin
UPMC Univ Paris 06, UR 2, L2E, F-75005, Paris, France
{guillaume.walck,michel.drouin}@upmc.fr

***Abstract*— This paper presents a complete 3D-reconstruction method optimized for online object modeling in the context of object grasping by a robot hand. The proposed solution is based on images captured by an eye-in-hand camera mounted on the robot arm and is an original combination of classical but simplified reconstruction methods. The different techniques used form a process that offers fast, progressive and reactive reconstruction of the object.**

I. Introduction

In health care and domestic applications, autonomous object grasping is a complex but required task for robots aimed at operating in human environments. Human grasping involves not only sensory motor abilities but also full knowledge of the object to grasp. In robot grasping, two different cases are generally considered : *i)* The object shape is fully recognized thanks to a preliminary learning process, then its position and orientation only remain to be determined. *ii)* The object has never been seen before, then its shape and dimensions are additionally to be characterized in order to grasp and manipulate it properly.

In manipulation applications with an anthropomorphic hand, a three-dimensional model of the unknown object will give the complete information that is necessary for grasp and task planning [1], [2]. Within the specific context of assistance to people, the robotic system should be responsive and capable of fulfilling the desired task in reactive time, which means in no more than a few seconds. Furthermore, as the system should operate in daily environments, the workspace is not supposed to be equipped with sensors. In this paper, we address therefore the problem of fast and reactive automatic 3D reconstruction in the particular case of unstructured environments.

To the best of our knowledge, previous works related to automatic fast 3D reconstruction with a single camera are sparse. The most similar work is [3], which proposed a real-time shape-from-motion reconstruction improved by visual servoing to observe the unknown scene. Such a technique can only recover models for basic primitive shapes.

Surveys on 3D reconstruction techniques are proposed in [4] and [5]. The compared methods especially issued from the computer vision community are slow, off-line and too outstanding for our grasping application. Moreover, to cover the scene, these solutions require either turn-tables or a camera network, often organized in a static setup. Both systems are not relevant to domestic places, which should not include image capture equipments. Therefore, to satisfy this constraint, a robot with embedded vision is needed. We consider a 6 degrees-of-freedom serial manipulator with an eye-in-hand camera mounted on its end-effector (see Fig. 1).

Our main purpose is to reconstruct the 3D model of an unknown object by looking at it during an observation trajectory using visual servoing. The trajectory generation and the camera pose estimation is not specifically the purpose of this paper (some details can be found in section III). It rather presents our work on a combination of simplified but efficient 3D reconstruction techniques. We only focus on the reconstruction process with an eye to a fully automatic modeling pipeline in the end. The proposed method is specially adapted for a sequential input of pictures captured with the single off-the-shelf calibrated camera from various angles of view along the observation trajectory. The medium detailed 3D models are computed on-line on a single computer in reactive time. A few millimeters accuracy is aimed within a less than a 15-second period.

The remaining of the paper is organized as follows. Section II presents previous works on 3D reconstruction classified in categories and highlights the usable methods for our application. Section III describes each step of our solution with details on adapted algorithms. Results for simulated views of virtual objects and captured views on our setup are analyzed in section IV. Section V concludes and proposes future work research.

II. Previous Work

Volumetric scene or object 3D reconstruction methods from multiple views have been largely developed over this past two decades and led to more and more precise models but often with no realistic computation time to be directly used in robotic manipulation or navigation problems. These methods may be split into two classes : surface-based methods and volumetric-based methods.

A. Surface-based methods

In this first class, reconstruction methods are traditionally based on image matching, including in particular recent

The research leading to these results has been partially supported by the HANDLE project, which has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT 231640

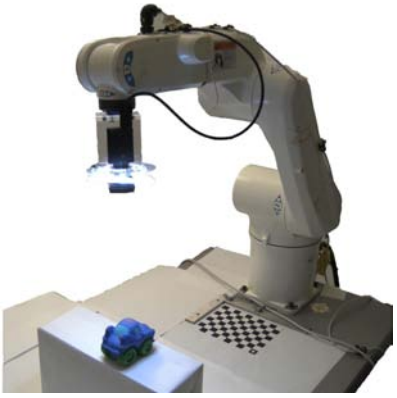

Fig. 1. 6 Degree-of-freedom manipulator with an eye-in-hand camera

multi-view stereo techniques [6]. They chiefly use correspondence techniques to obtain several local models often represented as *depth-maps* that must be fused into a single consistent model to reproduce the object envelope. Views must be close together, such that a great number of cameras is necessary. Furthermore, there is no easy handling of occlusions and visibility problems. Even if building a depth-map is faster nowadays, the fusion part implies a heavy computational cost [7], which hinders the use of such methods for our application.

Surface reconstruction by optimization algorithms based on *Level-sets* techniques [8], [9] is another way to progressively approximate the object envelope. An initial surface is iteratively deformed to minimize a two-component energy function mainly based on the photo-consistency of surface elements according to several views. Level-set methods inherently handle possible changes in topology, which is an advantage for multiple objects reconstruction. The value of the energy function must be defined in the entire space or extended from the surface, which is not always straightforward. The method also suffers from complex and expensive computations as it works iteratively on a huge space. Even hardware accelerated implementation of this solution on Graphical Processing Units (GPU) cannot decrease the computation time under 10 minutes [5]. Hernández et al. [10] combined silhouette and stereo information in a snake framework where different driven forces shrink a deformable surface model, unfortunately leading also to a few hours long reconstruction time.

B. Volumetric-based methods

The second class of methods consists in computing consistent surface or volume models directly in 3D space. They do not require a matching process and consequently only a low number of calibrated cameras may be used to save time. Those methods suppose a bounded area embedding the object to be known. Commonly, this surrounding box is divided in regular sufficiently small cubes called voxels that will be carved according to their consistency to the object. The consistency can be determined in two ways. The first way is a binary decision. Voxels whose projection lies in the silhouette of the object in each view produce an envelope called Visual Hull (VH) [11]. For many practical usages, these methods lead to accurate enough models, in near real time, but the object must have no concavity area, which limits the applications. The other way to check consistency is to use photometric information to improve the model in concave zones. *Voxel Coloring* [12], *Space-Carving* [13], and *Generalized Voxel-Coloring* [14] are the best known methods using photo-consistency. Occlusion and visibility problems can be taken into account in several techniques. However, Voxel-Coloring methods lay conditions on the camera locations to avoid the visibility problem, which are too restrictive for our application. The two others bypass this restriction and lead to relatively simple and highly-parallel algorithms but suffer from the fact that each voxel is tested independently of its neighbors, leading to non-smooth models with holes or floating voxels.

Another type of methods aims to compute the surface as the minimum of an energy function somewhat like level-set methods but starts from a discretized volume. All of those methods use the principle of *Graph-cuts* [15]. This reconstruction technique is based on the extraction of the surface separating two domains (inside/outside) through a cut in a graph equivalent to minimizing a criterion [16]. A possible graph structure is composed of nodes for each voxel, linked together with weights depending on the photo-consistency assorted with a smoothing term. The method provides high fidelity reconstruction models but is slow for dense graphs.
An active volumetric reconstruction has also been proposed in [17], inspired by the *Snakes* where a 3D deformable model is shrunk under the influence of several forces without using optimization procedures. It leads to reasonable computation times, about one minute for a 128^3 voxel box, but still too long for our application.

A few methods [18], [19] combine some of the mentioned techniques mainly for high-detailed objects but do not use their advantages for fast reconstruction. We focus in this paper on a combination of classical but simplified volumetric methods and adapt them for a progressive reconstruction process during image capture.

III. Combined Methods for Progressive Reconstruction

Our work aims at reconstructing a few-millimeters accuracy 3D model of an unknown object in reactive time – no more than a few seconds – using an eye-in-hand calibrated camera that takes a set of images during a servoed trajectory around this object. The whole object modeling pipeline will not be described here, this paper only presents the 3D reconstruction part (A,B,C in Fig. 2) based on views and silhouettes manually prepared but sequentially sent in the pipeline. However, we will evaluate variations in the quality of the reconstruction due to errors in silhouette extraction and camera pose estimation.

The reconstruction procedure is composed of three steps (A,B,C). The main idea is to use a coarse-to-fine approach where the first step A consists in progressively determining

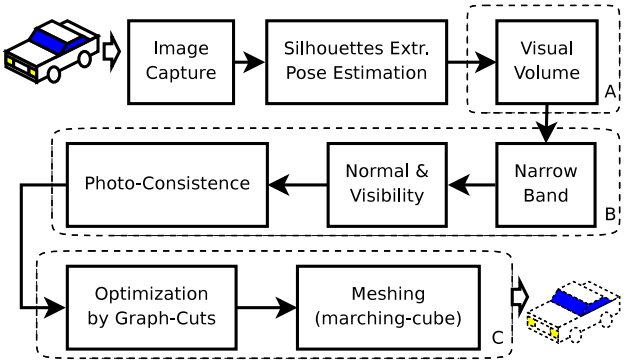

Fig. 2. Complete reconstruction pipeline

the VH. This step allows us to extract an initial coarse surface immediately at the end of the movement around the object. In most applications, this step is not sufficient. The concavity is never visible in the silhouettes and cannot be recovered. In order to improve the object model, additional information like photo-consistency has to be evaluated as it is done in Voxel Coloring. However, we choose here to pre-process the initial surface in a second step B, avoiding then visibility problems of such kind of methods and reducing the number of voxels to treat. We determine a band of voxels in which the initial surface lies. For each voxel, we estimate its visibility and the set of cameras to use for computing its photo-consistency. The last step C aims at finding one best estimation of the true object surface, contained in the band, using a graph-cut optimization technique with a limited number of nodes.

For multi-view reconstruction, image data is the main input to every process. As the picture capture method is not treated in this article and is an on-going work, only a few details are given here. An automatic coverage of the object surrounding domain by visual servoing is performed so that the object silhouette remains in the center of each captured picture. This observation movement covers many view points on a kind of front half-hemisphere automatically centered on the object. The experiments show that only 10 to 12 images equally distributed on the observation path are sufficient for the modeling. Furthermore, the visual servoing allows us to determine the location and size of a bounding box E containing the unknown object.

We assume that the pictures $I_1 \dots I_N$ ($N \leq 12$) provided to our reconstruction pipeline are calibrated (projection $\mathcal{P}_n$) using common eye-in-hand calibration methods [20] to extract end-effector to camera frame transformation. They also fulfill the standard Lambertian diffusion model on the object surface. Silhouettes $S_1 \dots S_N$ are supposed to be calculated as soon as an image is taken. For better evaluation purposes, the silhouettes are extracted manually but will be later processed automatically with an adapted *growcut* method [21]. Note that standard background subtraction is impossible during visual-servoed movements.

A. Progressive Visual Hull reconstruction

We compute a VH during the observation movement. Input silhouettes $S_1 \dots S_N$ are used to progressively carve the initial volume E. For each voxel $v \in E$, a square patch is created around the projection of its center $\mathcal{P}_1(v_{xyz})$ in the current view I_1. To avoid projecting 8 voxel corners at each time to estimate the patch size as done in [12], we approximate the patch size according to the voxel size and the projection factor. If at least one pixel in the patch belongs to the silhouette S_1, the voxel is part of the object and is then pushed in a linked-list L_1.

Further iterations $n = 2 \dots N$ proceed the same way, checking the pixels of the patch around $\mathcal{P}_n(v'_{xyz}), \forall v' \in L_{n-1}$, removing outliers from L_{n-1} to create L_n. Thanks to this sequential process, a next image I_{N+1} can be added to the pipeline and treated in a similar fashion based on L_n. The silhouette membership condition used leads to create a slightly bigger volume around the object true surface. In fact, this margin can even be advantageous since it helps to ignore possible artifacts on the silhouette contour. Processing through smaller linked-lists L_n at each iteration is fast and well adapted to a sequential capture as it progressively carves the volume, but the quality of the resulting model relies on a good silhouette extraction (see section IV for details).

The VH V in which the object is sure to lie, will be the base volume for the following pre-processing and optimization steps.

B. Pre-processing of the Visual Hull

In order to reduce the number of voxels to process, we use the principle of a narrow band, also called *crust*, as applied in [18]. Voxels on the border of the initial surface $\mathcal{S}_{base}$ are extracted from the VH V and denoted *InBand*. This label is spread to 2 inner layers inside the volume, up to a $\mathcal{S}_{in}$ surface where voxels are then labeled *InSide*. Outside $\mathcal{S}_{base}$, a layer is created and its voxels are labeled *OutSide*. Voxels in these three types of layers are the only one processed in the next steps and will be called *Huxels* for Hull Element. Linked-lists are used to store *huxel* objects and their properties (status, neighborhood, etc...).

Occlusion of some *huxels* by others must be taken into account to avoid false photo-consistency measurements, without constraining the camera placement. For this reason, we pre-process the *huxels* to estimate for each both their visibility and an approximate normal vector to the surface. In the first *InBand* layer, the normal vector is roughly computed as a sum of direction vectors oriented towards the *OutSide huxels* in the neighborhood (6-connected). As for the visibility, we rather determine the set of views where each *huxel* is seen. The knowledge of this set will help to select the right point of view for consistency calculations. Visibility is not computed according to the rough normal vector but using standard OpenGL rendering with Z-buffer as proposed in [22]. In each view, *InBand huxels* are rendered under the corresponding camera pose and parameters with *DepthTest* mode active so that only front *huxels* are seen. Drawing each *huxel* in a coded color allows to find which *huxel* is seen from the view and eventually build the complete set.

Again, to minimize the number of computations, we take

advantage of an extension possibility mentioned in [13]. If the band considered is small enough, each layer that is a superset of $\mathcal{S}_{base}$ shares similar parameters. Visibility and normals are then computed only for *huxels* on the first *InBand* layer and are propagated to both inside and outside layers by merging neighbor *huxels* data.

Now that normal vectors and visibility are known for each *huxel*, photometric data included in selected best views can provide photo-consistency information for the *huxel* set in order to refine the model.

Photo-consistency measurements are based on the hypothesis of homogeneous light diffusion. Sampling intensity or color values on the surface in several views respects this condition if the involved views are not far from the normal to the surface of the considered element.

Vogiatzis et al. [18] choose the participating views according to two angles. The angle between a pair of cameras and the one between the viewing direction and the normal to the surface. To simplify this method, for each huxel we choose to sort the list of available views from the set of views that see the *huxel*. We arrange the views according to the angle formed by the viewing direction and the normal vector and select the three first ones with the smallest angle. We reject views too far from the normal.

Sampling is done on a patch around the projection of the *huxel* center in each three views, and the photo-consistency measure is the average of a normalized cross-correlation on the patches for only 3 pairs of views. We used the same function as proposed by [18] with $\sigma = 0.8$ to map our photo-consistency term $\rho(v)$ between 0 and 1, zero when the *huxel* is consistent meaning theoretically that is is on the object surface. In case of a single view seeing a *huxel*, this *huxel* is marked not to be treated the same way. It keeps its original status when it is *InSide*/*OutSide* and swaps status to *InSide* when it is *InBand*. Fully pre-processed *huxels* can be treated in a final optimization step, revealing possible object concavities.

C. Fine model reconstruction

The final reconstruction step consists in finding the closest surface in the band using an optimization technique by graph-cuts.

The optimization technique requires to build a graph upon the *huxels* in the band. *Huxels* are converted into nodes but each layer (see III-B) of the band is inserted differently in the graph structure (Fig. 3) : *InSide huxels* must always keep their status and are therefore linked to the *Source* terminal with infinite weight. The same is done for *OutSide huxels* linking them to the *Sink* terminal with infinite weights in order to keep their status. Only *huxels* labeled *InBand* can change their status depending on the cut in the graph. They are linked to the *Source* with a smoothening term λ called *ballooning term*.

To insert the consistency in the energy to minimize, each node is linked to its 6-connected neighbors with a weight equivalent to the mean value of photo-consistency between

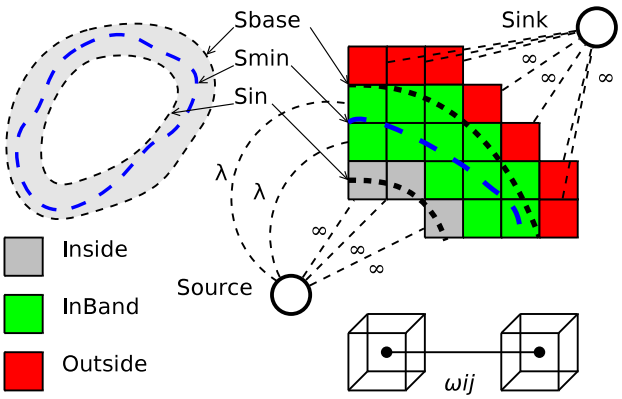

Fig. 3. Graph structure : *Inside* (resp. *Outside*) linked to *Source* node (resp. *Sink*) by infinite weight, *InBand* linked to *Source* node by constant λ (ballooning term) and neighbors linked with ω_{ij}

the pair of linked nodes i and j : $\omega_{ij} = \left(\frac{\rho(i)+\rho(j)}{2}\right)$. Uncertain *Huxels* marked to be treated differently have no neighborhood links but remain linked to the source/sink.

The cut in the graph separates the nodes into two parts, optimizing the energy function. Nodes still linked to the *Source* (resp. *Sink*) change their status to *InSide* (resp *OutSide*). The remaining *InSide huxels* added to inside voxels compose the optimal volume we are looking for.

The final volume provides information about the shape and the size of the object. If needed, a meshing step can be added to the process in order to create a polygonized surface of the resulting voxelized volume. We apply the marching-cube technique [23] often used in medical imagery to extract iso-level surfaces. This method is fast and systematic, and easily extracts the surface between interior and exterior voxels. This rough surface will be used for comparison with ground truth surface.

IV. Experiment & Results

A. Experiment & Measurement methods

To test our algorithms, we use both virtual views of synthetic objects and real captures from our eye-in-hand camera (Fig. 4 & 7). Ground-truth is not available for our real objects but the virtual objects are fully known and their existing model will serve the purpose of quality measurements.
Lighting conditions, surface diffusion and camera calibration are fully controlled for the synthetic object. This is not the case for real objects. However, calibration of the real system is done only once to extract the camera parameters and its relative position to the end-effector. The precise position of the robot arm end-effector gives the extrinsic parameters during the observation movement. Moreover, rather good light conditions were obtained by embedding an annular light on the camera.
The complete process was executed on a PC with a 2.8 GHz PentiumIV processor and 1 GB memory. As the VH is due to be extracted during the movement around the object, the time for this step will mainly depend on the speed of the visual servoing trajectory, hiding the time for VH construction. Therefore, for the overall process time measurements only, the input images are captured and prepared

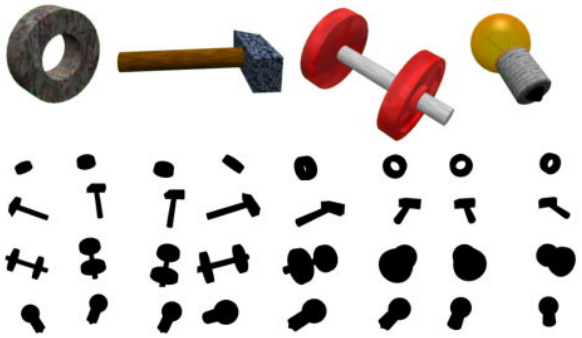

Fig. 4. Synthetic objects (Roll, Hammer, Weights, Light Bulb) and part of their silhouettes extracted from 12 views on a half-hemisphere

separately but are sequentially provided to the reconstruction pipeline. To evaluate the reconstruction quality, we consider the distance between the polygonized surface and the ground-truth surface of the synthetic objects. We use a Hausdorff distance between 2 closest triangles of the compared surfaces as proposed in [24]. The software they provide gives this distance in percentage of the bounding-box diagonal (BBD).

B. Result & Analysis

Accuracy: Reconstructions of synthetic objects were compared to the ground-truth in Fig. 5. The maximum error (darker areas) appears in zones where visibility is bad or null and in deep concavities. Due to marching-cube meshing, the surface does not seem visually smooth but the reconstruction is accurate enough : for a synthetic object whose BBD size is between 10 and 20 centimeters, a mean error of 1 percent of the BBD leads to a 1 or 2 millimeters error (Tab. I), which is precise enough for grasping.

Computation Time: Each step of the process has been timed, and the results are synthesized in Tab. I. Implementing the algorithm with progressive computations leads to a computation time shorter than 20 seconds. Visibility evaluation is the longest step of the process, due to the precision needed for a high-fidelity estimation. The large difference in reconstruction time for some objects comes from the size of the initial bounding-box and the occupancy ratio (number of voxels in V / Grid size). Thinner objects are quicker to reconstruct for instance.

Space coverage: We evaluated the influence of the number of views and their placement on the VH extraction and on the overall reconstruction quality for a synthetic cube

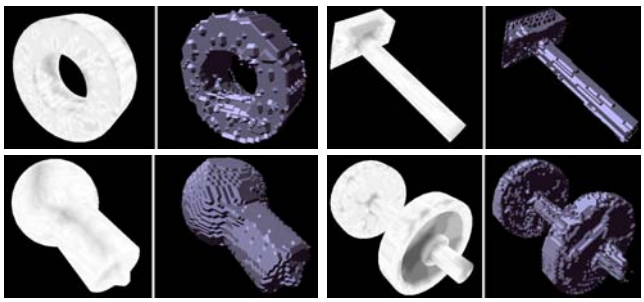

Fig. 5. For each virtual object : Left, quality measure errors range from black (maximum) to white (minimum) and right, meshed object reconstruction are rendered

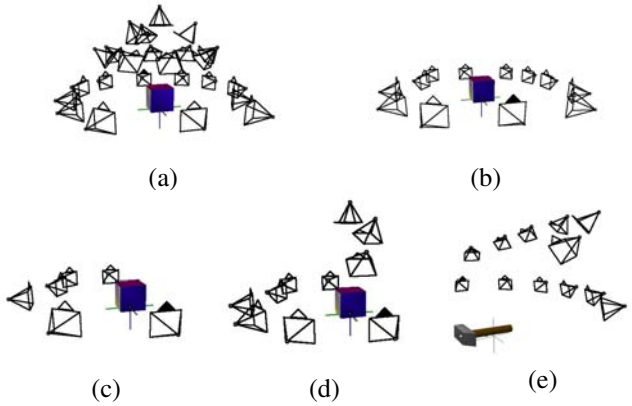

Fig. 6. Object coverage with 24 views in a hemisphere (a), 12 views in circle (b), 6 views in a half-circle (c), 10 views along a possible eye-in-hand camera trajectory (d) and 12 views in a half-hemisphere (e)

(Fig. 6). Analysis of the mean error on the cube VH and full reconstruction (see Tab. II) shows that the mean error does not dramatically increase with a reduced number of views, even for a front half circle with 6 views. Parts behind the objects only are not well reconstructed in this case. Influence of the lack of details behind modelized objects will be assessed according to grasping techniques in a future work but a primary test with a grasp configuration generator [25] is encouraging. We concluded that 10 image captures during a movement along 2 perpendicular arcs (Fig.6 (d)) can provide enough data for our application.

Robusteness: Because VH construction and consistency estimation are strongly related to precise silhouettes extraction and camera pose and parameters, we analyzed their influence on our reconstruction quality.

Incorrect view placements are deviated from perfect positions by adding random errors. A $\pm 1mm$ error on the positions mostly affects the consistency estimation but its influence depends on the projection factor. The Roll mean surface error reaches 2.65% of BBD compared to 1.86% whereas it only changes from 1.10% of BBD to 1.33% for the Hammer. Indeed, the Roll is closer from the camera than the

TABLE I

ACCURACY AND PROCESSING TIME

Objects	**Roll**	**Hammer**	**Weights**	**Light Bulb**
Grid size	50x30x60	50x90x30	54x100x55	50x65x65
Voxels $\in V$	18125	24405	63367	46049
Computation time for each reconstruction step (s)				
Visual Hull	1.28	1.55	3.52	2.59
Band creation	1.01	1.57	3.45	2.17
Visib. & Norm.	4.52	4.63	9.36	5.07
Photo-Consist.	0.47	0.6	1.15	0.65
Optimization	0.16	0.18	0.35	0.18
Total time	**7.44**	**8.53**	**17.83**	**10.66**
Reconstruction error (% of the BBD)				
Mean error	1.86	1.10	1.18	0.64

TABLE II

INFLUENCE OF OBJECT COVERAGE ON RECONSTRUCTION QUALITY

Coverage type	**(a)**	**(b)**	**(c)**	**(d)**
Mean error (% BBox diag.)				
for VH only	1.22	1.59	1.91	**1.50**
for full reconstruction	0.94	1.11	1.69	**1.16**

Fig. 7. Camera view and reconstructed model for real objects

Hammer is and the projection factor for the Roll gives a ratio of 4.5 pixels for 1mm in 3D space that must be compared to our 7 pixels patch size for consistency check.

The quality of the perfect silhouettes is also degraded for evaluation by adding holes or spots in the binary masks. Spots do not influence the VH reconstruction since wrong voxels are generally not consistent in each view and thus disappear during the process. The contrary is not true as a single hole in the silhouette means carving good voxels that are definitely lost for further iterations. Using entirely filled silhouettes is mandatory but such a quality could be reached in complex backgrounds with a growcut method [21].

Real Objects: Early reconstruction results for real objects captured on our experimental setup are presented in Figure 7. The automatic observation movement provided 10 images in 20 seconds but the VH was progressively calculated in less than 13 seconds. Models composed of 200,000 voxels with millimetric resolution were obtained within 18 to 35 seconds after the movement. Due to remaining calibration errors, the optimization step did not refine the model as effectively as for virtual objects. Work is going on to improve both the pre-processing and optimization part mainly by switching to GPU-based computation when possible in order to get the complete process running in less than 20 seconds.

V. Conclusion & Future Work

In this article we have proposed a complete and fast 3D reconstruction method for unknown objects providing results in reactive time. Starting from 10 images captured by a eye-in-hand system, our process is able to progressively modelize the object during a movement around it with millimetric accuracy, which is sufficient for grasping tasks. We have presented an original combination of methods and simplified the computations for our requirements without important quality loss. Working on interesting areas of the object only and processing images sequentially help to reduce the computation costs.

Our future work will be devoted first to silhouette extraction to improve robustness in unknown environment (non-homogeneous background). A Growcut method [21] combined with active graph-cut techniques [26] for region-based segmentation seems the most adapted to fast but sequential image processing. Another aspect that will be considered concerns the computation time. Converting most of the parallel processing on GPU should accelerate the reconstruction.

References

[1] J.-P. Saut, A. Sahbani, S. E. Khoury, and V. Perdereau, "Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces," in *IROS 2007*, 2007, pp. 2907–2912.

[2] C. Michel, C. Remond, V. Perdereau, and M. Drouin, "A robotic grasping planner based on the natural grasping axis," in *Int. Conference on Intelligent Manipulation and Grasping*, 2004.

[3] E. Marchand, F. Chaumette, and E. Rutten, "Real time actual visual reconstruction using the synchronous paradigm," in *IROS'95*, 1995, pp. 96–102.

[4] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, pp. 7–42, 2002.

[5] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *CVPR*, June 2006, pp. 519–528.

[6] M. Goesele, B. Curless, and S. M. Seitz, "Multi-view stereo revisited," in *CVPR*, 2006, pp. 2402–2409.

[7] M. Sormann, C. Zach, J. Bauer, K. F. Karner, and H. Bischof, "Watertight multi-view reconstruction based on volumetric graph-cuts," in *SCIA*, 2007, pp. 393–402.

[8] O. Faugeras and R. Keriven, "Variational principles, surface evolution, pdes, level set methods and the stereo problem," *IEEE Transactions on Image Processing*, vol. 7, pp. 336–344, 1998.

[9] H. Jin, S. Soatto, and A. J. Yezzi, "Multi-view stereo reconstruction of dense shape and complex appearance," *The International Journal of Computer Vision*, vol. 63, pp. 175–189, 2005.

[10] C. Hernández and F. Schmitt, "A snake approach for high quality image-based 3d object modeling," in *2nd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision, Nice, France*, October 2003, pp. 241–248.

[11] A. Laurentini, "The visual hull of curved objects," in *ICCV (1)*, 1999, pp. 356–361.

[12] S. Seitz and C. Dyer, "Photorealistic scene reconstruction by voxel coloring," *Int. Journal of Computer Vision*, vol. 25, no. 3, 1999.

[13] K. N. Kutulakos and S. M. Seitz, "A theory of shape by space carving," *International Journal of Computer Vision*, vol. 38, pp. 307–314, 2000.

[14] W. B. Culbertson, T. Malzbender, and G. G. Slabaugh, "Generalized voxel coloring," in *ICCV '99 : Proceedings of the International Workshop on Vision Algorithms*, London, UK, 2000, pp. 100–115.

[15] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[16] Y. Boykov and V. Kolmogorov, "Computing geodesics and minimal surfaces via graph cuts," in *ICCV*, 2003.

[17] X. Liu, H. Yao, Y. Chen, and W. Gao, "An active volumetric model for 3d reconstruction," *IEEE International Conference on Image Processing (ICIP 2005)*, vol. 2, pp. II–101–4, Sept. 2005.

[18] G. Vogiatzis, P. Torr, and R. Cipolla, "Multi-view stereo via volumetric graph-cuts," in *CVPR*, vol. 2, 2005, pp. 391–398.

[19] A. Hornung and L. Kobbelt, "Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding," in *CVPR*, vol. 1, 2006, pp. 503–510.

[20] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *IROS 2006, Beijing, China*, 2006, pp. 4647–4653.

[21] V. Vezhnevets and V. Konouchine, ""grow-cut" - interactive multi-label n-d image segmentation," in *GraphiCon*, 2005, pp. 150–156.

[22] P. Debevec, Y. Yu, and G. Borshukov, "Efficient view-dependent image-based rendering with projective texture-mapping," in *9th Eurographics Rendering Workshop*, 1998, pp. 105–116.

[23] W. E. Lorensen and H. E. Cline, "Marching cubes : A high resolution 3D surface construction algorithm," in *Proceedings of ACM SIGGRAPH'87*, vol. 21, 1987, pp. 163–169.

[24] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "Mesh : Measuring errors between surfaces using the hausdorff distance," in *Int. Conf. on Multimedia and Expo*, vol. I, 2002, pp. 705–708.

[25] S. El-Khoury, "A hybrid approach, analytical and empirical, for natural grasps synthesis with a multi-fingered robotic hand," Ph.D. dissertation, UPMC, 2008.

[26] H. Lombaert, Y. Sun, L. Grady, and C. Xu, "A multilevel banded graph cuts method for fast image segmentation," in *Proceedings of ICCV'05*, vol. 1, 2005, pp. 259–265.